

Arhitectura Intel Core i7

Ioan Deaconu

Automatic Control and Computers Faculty
University Politehnica of Bucharest,
{ioan.deaconu@cti.pub.ro}

***Index Terms*—Intel Core i7, Nehalem**

I. INTRODUCERE

Intel, pe langa AMD, este unul din liderii mondiali ai industriei de microprocesoare. Acest lucru este datorat in mare parte arhitecturii Core i7, arhitectura revolutionara lansata in noiembrie 2008.

Desi de atunci Intel a ajuns la a 5-ea iteratie a arhitecturii, sub nume de cod Broadwell, functionarea ei este identica cu cea a arhitecturii Nehalem, prima iteratie a lui Core i7.

Din acest motiv, aceasta lucrare va prezenta imbunatatirile aduse acestei arhitecturi, imbunatari care chiar si dupa 7 ani de la introducerea lor sunt inca folosite.

II. NOTIUNI TEORETICE AFERENTE

Specificatiile de baza sunt:

- Hyper-threading.
- Quad-core si octa-core nativ
- Intel QPI
- 64KB L1 cache pentru fiecare nucleu (32KB L1 data si 32KB L1 instruction)
- 256KB L2 cache pentru fiecare nucleu
- L3 cache de ordinul MB
- Controler de memorie ram DDR3 integrat cu doua sau trei canale
- A doua generatie de Intel-VT
- Instructiunea SSE4.2
- Pipeline in 20 sau 24 de stagii.

Aceasta arhitectura are multe imbunatatiri care o ajuta sa fie relevanta si in aceste zile, dar voi prezenta cele mai importante dintre ele.

- Accesul nealiniat la memoria cache. Pana la arhitectura Nehalem, accesul la memoria cache era facut aliniat, astfel compilatoarele trebuiau sa se ocupe de alinierea memoriei pentru a obtine performanta maxima. Cu acces nealiniat la memoria cache, aplicatiile nu mai trebuie sa fie optimizate pentru accesul nealiniat, marind viteza de executie a programului[?].
- Hyper-Threading. Desi acest feature a fost folosit anterior in arhitectura Pentium 4 in anul 2003, s-a renuntat la el in urmatoarea generatie, pentru a se reveni inapoi la simultaneous threading odata cu arhitectura Core i7.

- Controlul Avansat al Consumului. Pe langa nuclee propriu-zise ale procesorului, exista un nucleu mai mic, un nucleu care se ocupa de monitorizarea parametrilor de functionare, parametri precum temperatura, consumul, optimizand puterea totala consumata. Astfel, acest micro-controler poate opri complet un nucleu pentru a minimiza consumul.
- Modul Turbo Boost. Datorita controlului avansat de consum, si a faptului ca mai multe nuclee pot fi oprite, aceasta rezerva de putere poate fi folosita pentru a mari frecventa nucleelor active. Cat timp puterea totala consumata a procesorului si temperatura acestuia nu depasesc limitele admise, exista posibilitatea ca toate nucleele sa poata functiona cu o frecventa marita in acelasi timp.

O mare parte din aceste idei au fost folosite in alte arhitecturi mai recente, arhitecturi precum Cortex A7, A9 sau mai recent A57 si A53. Deasemenea si AMD a folosit o parte din idei in procesoare x86 create de ei.

III. IMPLEMENTARE

A. Accesul nealiniat la memoria cache

Cea mai mare instructiune de tip SSE are 16 bytes, sau 128 de biti. Pentru fiecare operatie de tip store/load exista 2 versiuni: prima versiune care are operatorii aliniali la 16 bytes, si cea de-a doua versiunea care nu are operatorii aliniali.

Compilatoarele folosesc operatiile nealinitate daca nu pot garanta accesul aliniat la memorie. Toate procesoarele de generatie anterioara, Core 2, aveau o scadere drastica a performantei in momentul folosirii instructiunilor nealiniat fata de cele aliniat.

Problema este ca majoritatea compilatoarelor nu pot garanta alinierea datelor in memorie si astfel se folosesc foarte multe instructiuni nealiniat. Incepand cu arhitectura Nehalem, nu mai exista scadere de performanta datorata folosirii instructiunilor pe date nealiniat. Astfel compilatoarele nu vor mai fi nevoite sa alinizeze datele in memorie de teama scaderii vitezei de executie [?].

B. Hyper-Threading

Accesul nealiniat la memorie a fost necesar deoarece in aceiasi arhitectura exista simultaneous multithreading.

In mod normal, un procesorul nu are folosite toate modulele componente in acelasi timp, astfel ca simultaneous multi-threading foloseste aceasta idee prin simularea a 2 nuclee virtuale folosind un singur nucleu. Pentru acest lucru, a fost necesara marirea numarului de registrii de load de la 32 la 48 si numarul de registrii de store de la 20 la 32.

Exista arhitecturi in care se un nucleu fizic poate simula 4 nuclee virtuale, dar in cazul lui Core i7, Intel a decis sa foloasca doar 2 nuclee virtuale pentru fiecare nucleu fizic. Astfel in cazul in care pe un nucleu s-ar executa operatii cu intregi, unitatea in virgula mobilă, ar putea fi utilizata de un alt thread care ar dori sa foloasca operatii cu numere reale [?].

C. Controlul Avansat al Consumului

Intel Core i7 are 4 stari principale ale consumului[?].

- P-State - starea de performanta maxima a microprocesorului
- T-State - starea de sleep in care microprocesorul are frecventa redusa
- C-State - starea in care procesorul este in sleep

- S-State - starea de sleep ale sistemului

D. Modul Turbo Boost

Datorita existentei unui control dedicat al puterii, nucleele pot rula la frecvente independente. In cazul in care doar 2 nuclee din 4 sunt folosite, atunci cele 2 nuclee vor putea rula la o frecventa mai mare decat cea de baza , deoarece vor folosi puterea salvata prin inchiderea celorlalte 2 nuclee. De exemplu, daca frecventa de baza a unui procesor cu 4 nuclee este de 2,66 ghz atunci daca maxim 2 nuclee sunt folosite, acestea vor putea rula la o frecventa maxima de 3 ghz.

[?].

IV. SIMULARE

Capitolul asta e cu simulare, ceva

V. CONCLUZII

Concluzii ?

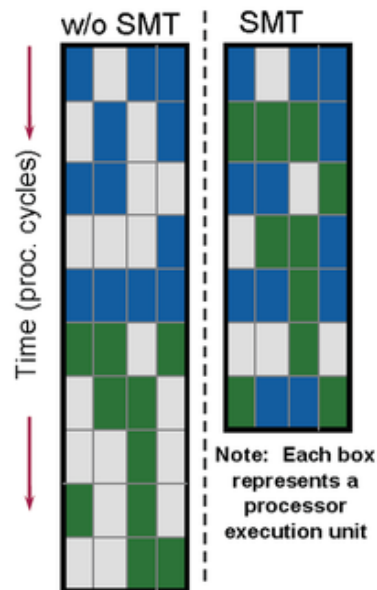


Figure 1. Diferenta intre cu si fara Hyper-Threading

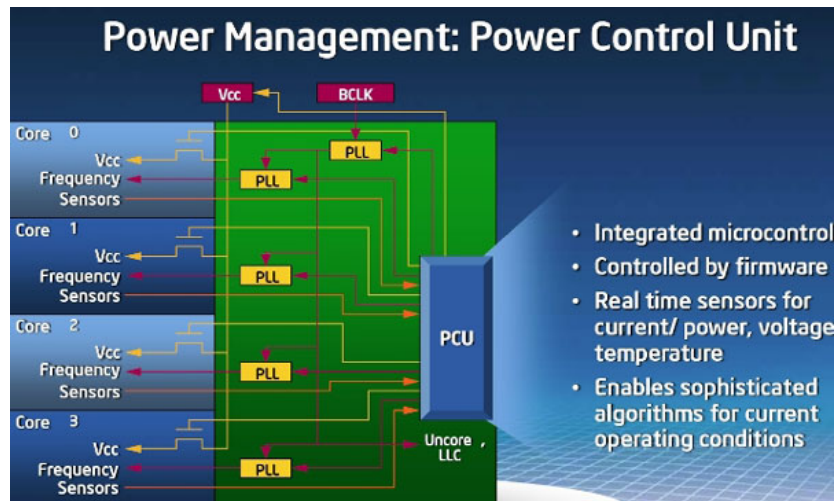


Figure 2. Unitatea de control a puterii

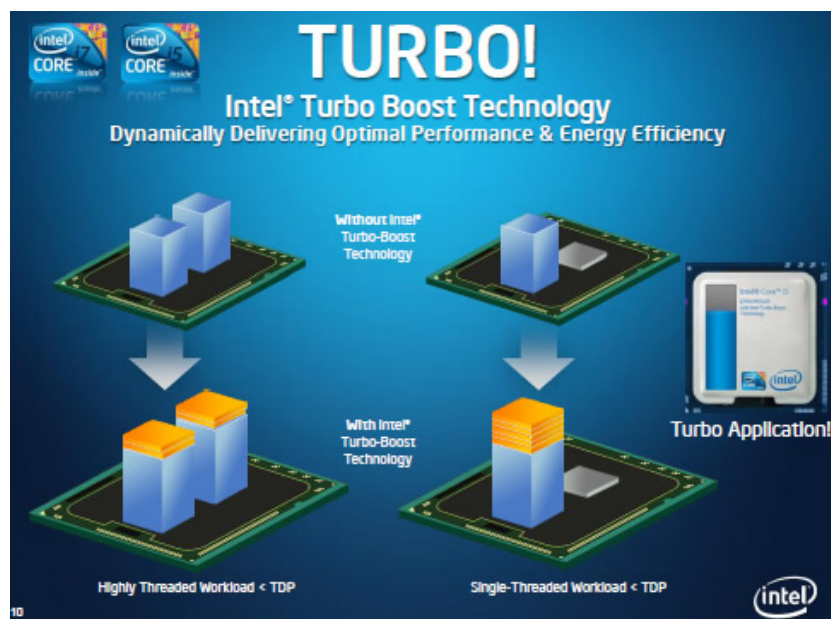


Figure 3. Exemple de Turbo Boost